Some Things You Always Needed to Know About Systems Engineering but Didn't Know You Needed to Know

Schneebaum Colloquium

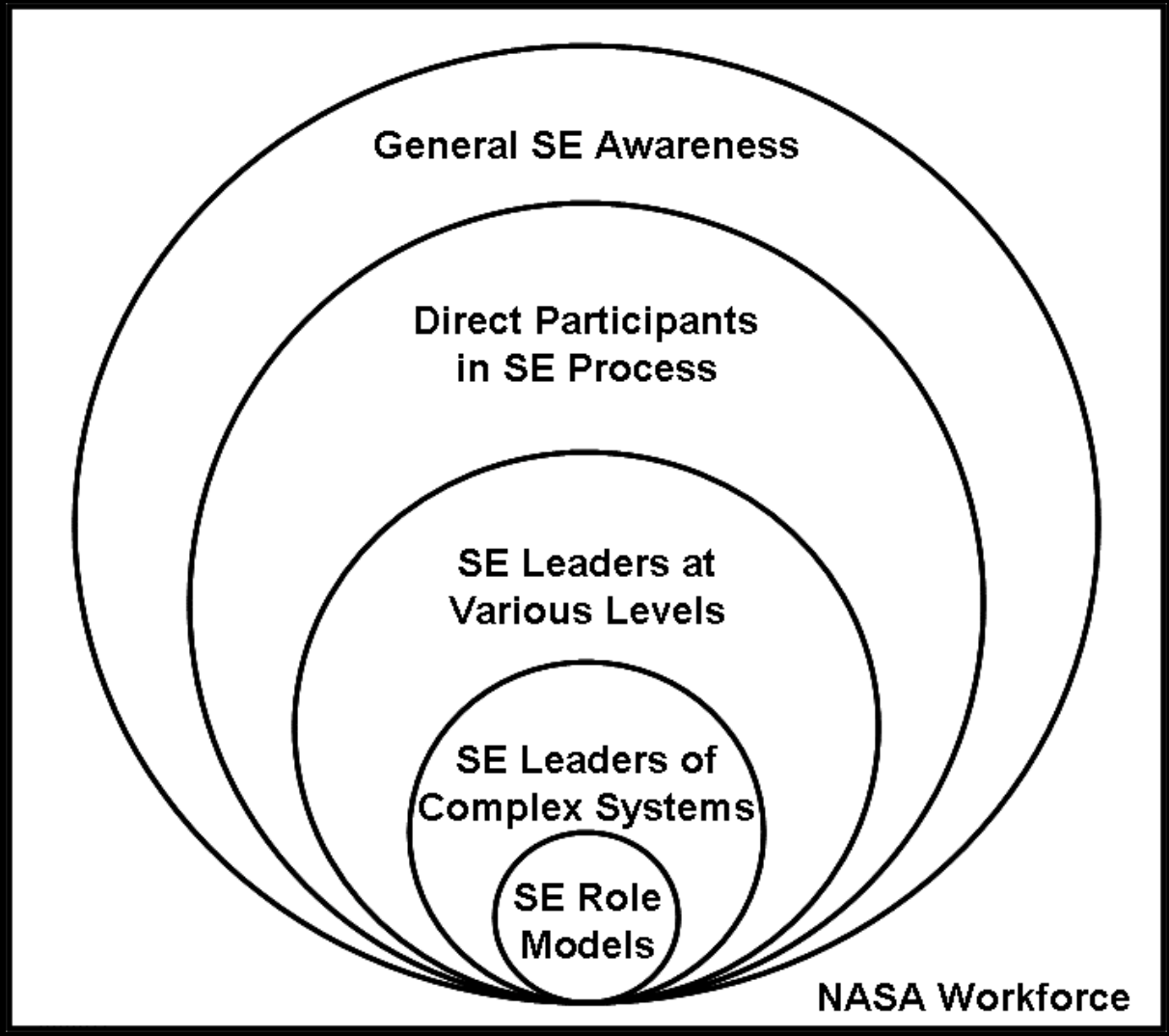
GSFC, September 2008

M.G. Ryschkewitsch

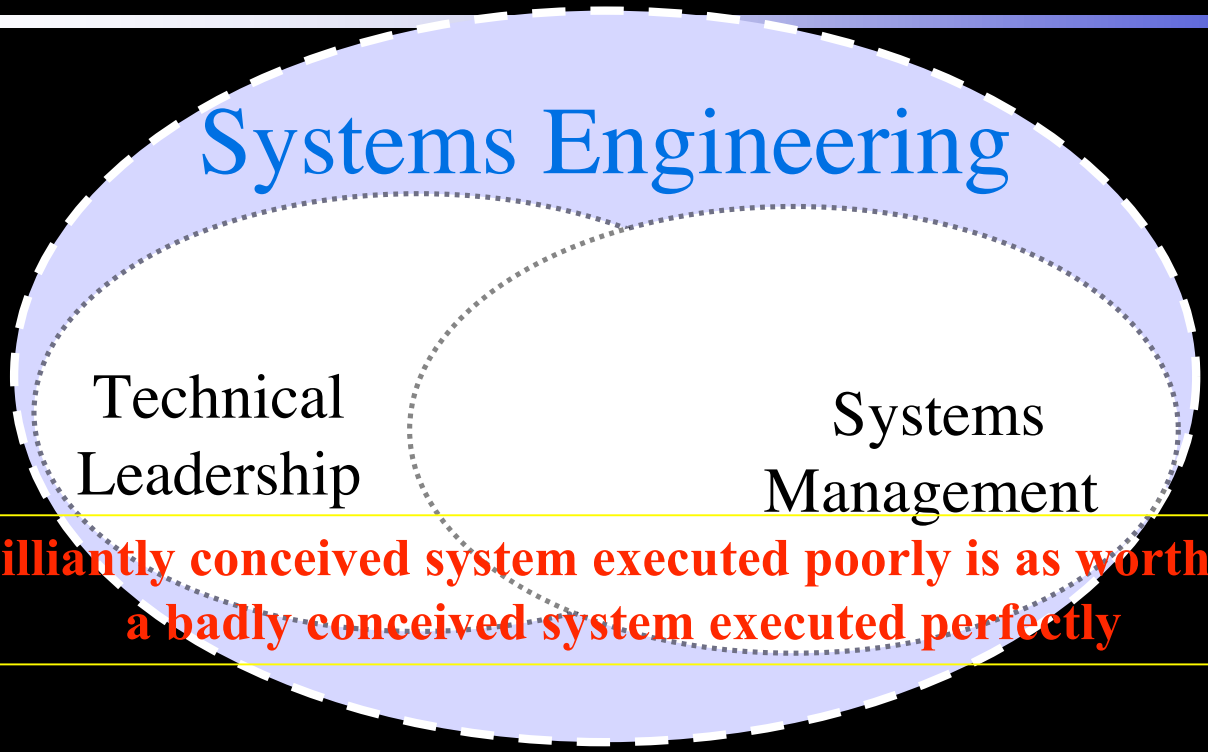
GSFC Alumnus

(& NASA Chief Engineer)

There are many levels of engagement and performance



So what is Systems Engineering Anyway?



A brilliantly conceived system executed poorly is as worthless as a badly conceived system executed perfectly

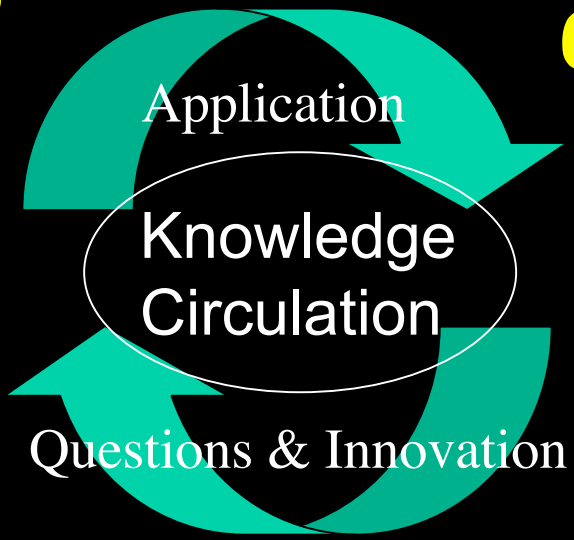
Creative Architect
Problem finder
Constructive Paranoid
Big Picture Thinker

Manager
Organizer
Process Controller
Detailed Verifier

There is no cookbook which can guarantee success but there are necessary ingredients

Completeness

Information, tools, techniques



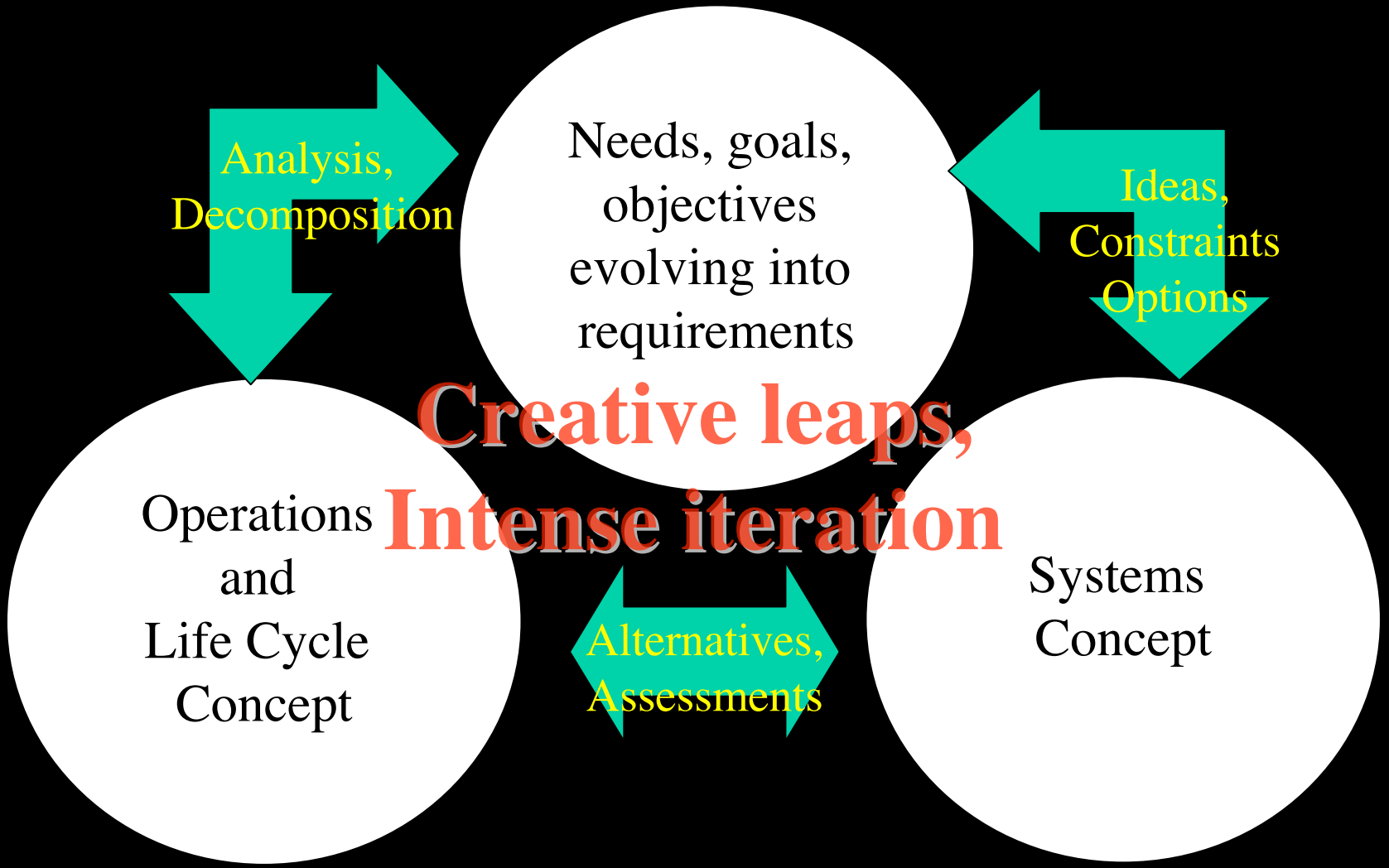
Consistency

Policy, Process

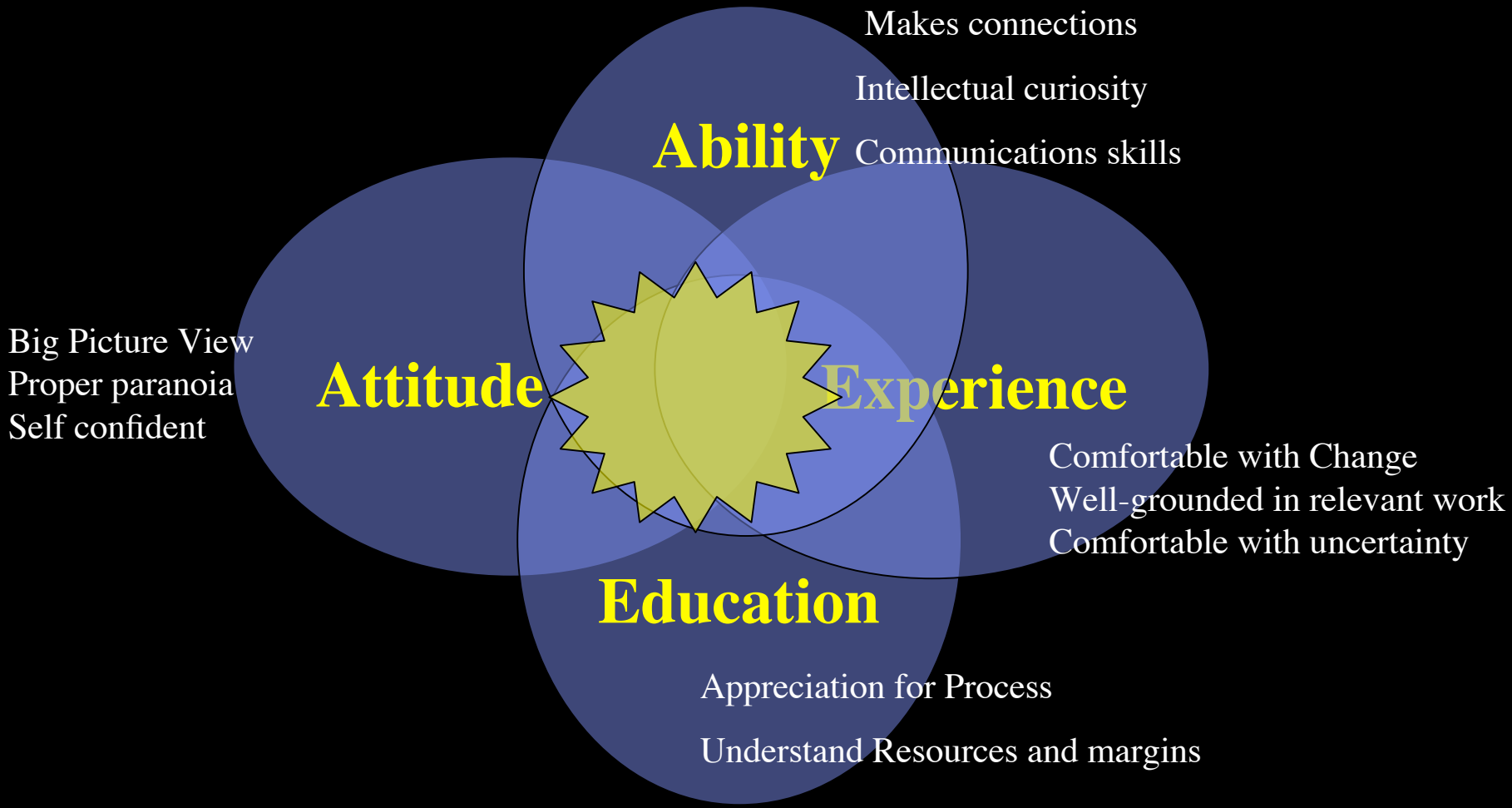
Creativity

Competencies
Human Behaviors
Culture of rigorous inquiry

Creation of new systems requires engineering science, art and lots of perspiration



A Systems Engineer requires a variety of capabilities- some can be learned in class, others are personal and must be developed



Leadership is essential!

"A team is made up of many individuals. The more individualistic, the better. When putting a team together, the manager should not try to find people whom he can outsmart, but people who are smarter than he is.."

"Then comes a test of leadership. All the people around the table are experts in their own field. Each one should be a strong individual, with strong feelings, capable of thinking problems through on his own -- or he should not be there. When a conflict arises, the director must be able to find a compromise solution that is best for the satisfactory accomplishment of the mission, and get willing agreement from the dissenters."

The Price of not asking “why”

All requirements sets are wrong, some are useful
(with apologies to George Box)

- Working to requirements is essential for good engineering but they must continually be validated
  - If we ever stop asking "Does this makes sense" we are poised for failure
- There are no such things as concept or operations independent requirements; One must always play the requirements and evolving design against the operations planning
- English is a lousy engineering language
- Analyses and models are always approximations

A systems engineer doesn't get paid for doing trades, she gets paid for deciding which trades to do

”Success is not simply the absence of failure, it also masks potential modes of failure” -- Henry Petroski

* Complex systems can fail catastrophically in ways which are impossible to predict

* We must allow for both the known unknown and the unknown unknown

* In the space business, no two systems can be the same; heritage is imperfect at best and usually less than we think

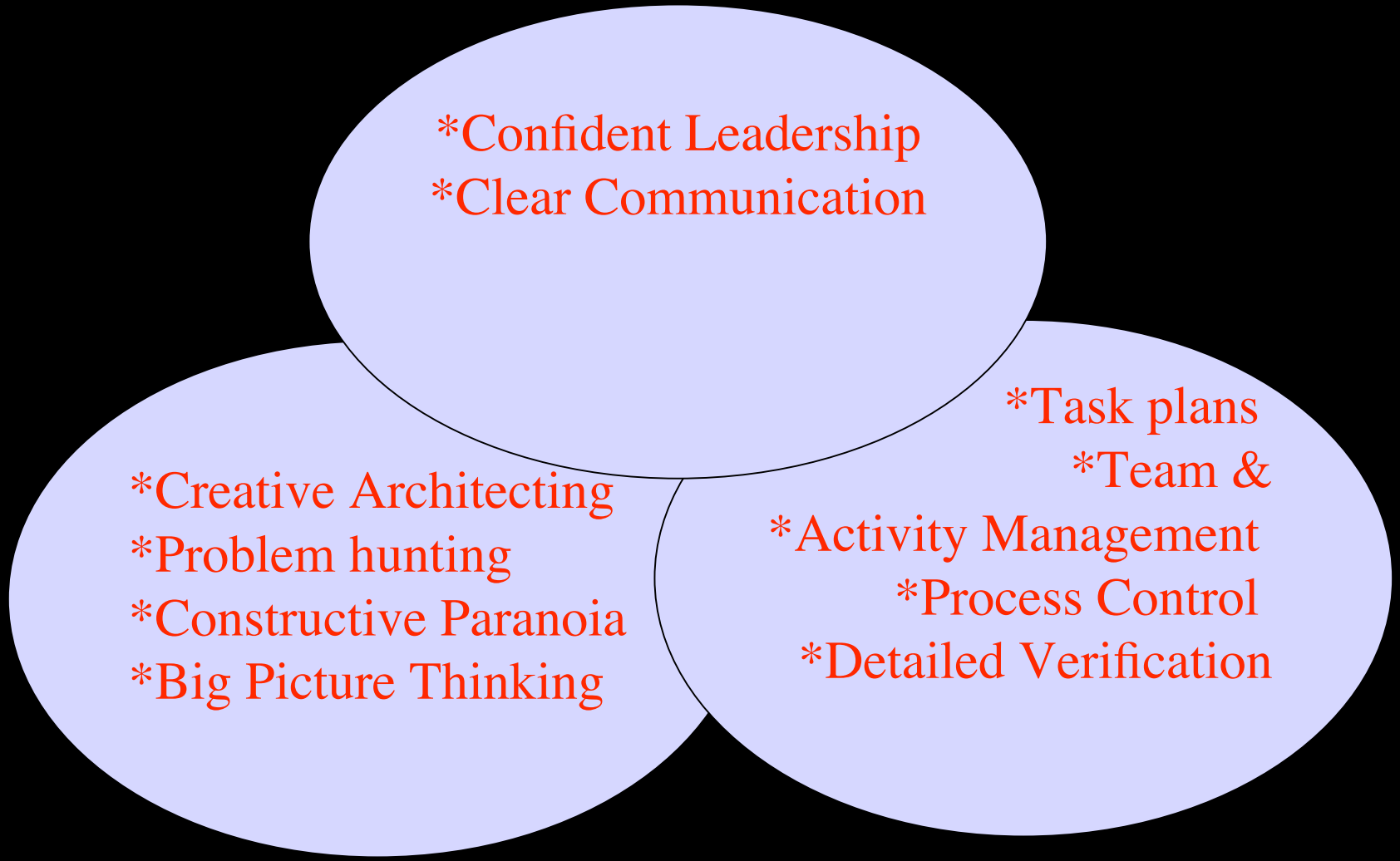
“test like you will fly” isn’t possible --How good are our tests and what do they NOT tell us -

*How much capability do we have beyond the requirement and how can we use that when things change

* What is the simplest possible “fail to” configuration which allows us to survive and fight another day

We must design for success by designing for faults and failures

Complete Systems Engineering is Key to Success



Some Suggested reading

----Mike Griffin, speech on the two cultures of engineering at Purdue University, March 28, 2007
http://www.nasa.gov/news/speeches/admin/mg_speech_collection_archive_2.html

----Ferguson, E.S. (1992) Engineering in the Mind's Eye. Cambridge, MA: MIT Press.

----Henry Petroski
- "Success through Failure: The Paradox of Design", Princeton University Press, 2006
- "To Engineer is Human", Vintage Books, 1992